An investigation into compositional features and feature merging for maximum entropy-based parse selection

Mullen, Anthony James

Chapter 8

Conclusion

8.1 Summation

This thesis has been concerned with approaches to selecting parses using the maximum entropy modeling technique. In particular, a novel approach to reducing noise in models and enhancing their generality is proposed and investigated. Experiments have been conducted in several quite different environments in this connection. Firstly, experiments were conducted in ranking sets of complete parses created by the Alvey Natural Language Tools grammar, reported in chapter 6. These experiments yielded positive results in the case where overfitting and sparse data presented the greatest problems. These experiments, in themselves, however, were not sufficient to demonstrate a significant practical advantage to using the feature merging technique. The model which showed significant improvement used highly specific features and was trained on a very small set of training data. Increasing by a factor of five the training data size eliminated any potential advantage of using the merging technique, and enriching the feature set, although increasing the size of the model, yielded much better results which could not be improved upon even with a commonly used frequency-based feature cutoff.

The investigation into employing an analogous strategy in the Alpino environment was motivated by the relatively small hand-parsed Dutch training sets available for Alpino. Since the Alvey Tools experiments indicated that insufficient training data was one of the criteria for a model benefiting from the merging approach, it seemed that the Alpino environment was in a good place to test it more conclusively.

Preliminary experiments on held-out data showed some promise, but this promise was not borne out in either of two experiments with different feature sets on a larger test suite of unseen data. The results of the Alpino experiments showed an increase in performance as a result of a frequency-based feature cutoff (a well-established method of combating overfitting) but no further increase in

performance with the introduction of merging prior to the cutoff, as would have been hoped.

8.2 Discussion

Toy examples such as the one presented in section 5.5 suggested that in certain cases, the interaction between a frequency-based feature cutoff and a merge threshold should yield desirable results, as opposed to a feature cutoff alone. The somewhat artificial task described in 6 demonstrated this further. However, in general, subsequent experiments in the working parsing environment of Alpino indicated that no advantage was gained by merging. What is it about the earlier case which allowed it to benefit from merging? What is it about subsequent cases that prevented the merging from helping?

One clue can be found in the subsequent experiments of chapter 6, featuring a larger set of training data and richer feature sets. The feature set of the first experiment reported was composed entirely of specific, partially lexicalized (prepositions and verbs) features. With a very small set of training data, overfitting was a major problem; the feature cutoff brought about significant improvement and incorporating merging allowed for features which contributed to modeling to be retained by bringing counts up above the cutoff. In short, for this model, the merging technique worked for precisely the reasons predicted in section 5.5.

However, the advantages gained by employing the technique decrease significantly as the task bears less resemblance to the toy example given in that section. In the Alpino experiments, the intention was to perform as well as possible, rather than demonstrate the potential for improvement. Thus the richest possible feature set was used. This feature set, as discussed in chapter 7 already included a variety of non-specific feature types, which would diminish greatly the positive impact of the generalized features created by merging, since the already-existing non-specific feature types would largely subsume the new features. Such non-specific features included rule features and subcategorization frame features. The rule features, it appears, were especially important to the modeling.

Revisiting the example from section 5.5, consider the feature set in figure 8.1, identical to the feature set in 5.5, except for the addition of a new feature, where we consider the "..." symbol to be an unspecified node which can be instantiated by any element.

Already, the problem posed by the cutoff is eliminated, since although the cutoff will discard features 3 and 4, instances of either of those features will also be instances of feature 2, so the total frequency count of feature 2 will be above the cutoff.

In the present toy example, the effect is identical to the use of merging. Given an original candidate feature set as in figure 8.1, then the features 2 and 3 in 8.2 have the same denotation. However, consider a feature set such as that in 8.3

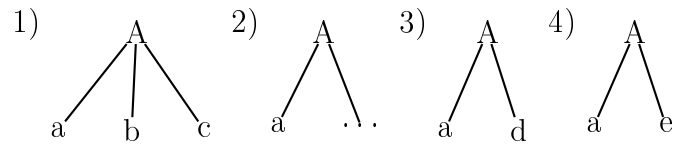

Figure 8.1: A toy feature set with a backed-off feature

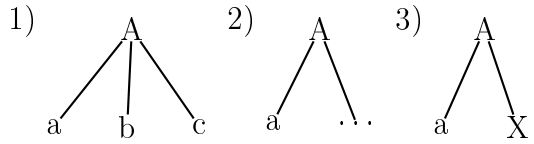

Figure 8.2: The feature set with the merged feature and the backed-off feature. Feature 2 and feature 3 have the same denotation, assuming an initial candidate feature set as presented in figure 8.1.

where another feature is introduced with the same basic structure as 2, yet with a feature "f". We imagine a case in which the frequency of "f" in the data is above the merge threshold and the frequency of the feature 5, containing "f" is above the cutoff threshold. In this case, the resultant feature set after the merge, as seen in 8.4, contains both 2 and 3, feature 2 being the original backed-off feature, and 3 being the merged feature.

In this case, however, the denotations of the two features are not identical. That is to say, feature 4, which includes element "f" is subsumed by feature 2, whereas it is not included in 3, which is strictly the union of 3 and 4 from figure 8.3.

The merged feature is a legitimate generalization of the two merged features. In the absence of better generalizations, it would contribute positively to modeling. However, in the feature set in figure 8.4, the backed-off figure 2 has important advantages which increase its relative importance, and consequently decrease the importance of the merged feature. First of all, the generalization is over *all* features which it subsumes, whereas the merged feature is a generalization only over an arbitrary subset of those features, namely the subset composed of the union of the merged features. The meaning of such a generalization much less clear than that of a simply backed-off feature. Capturing meaningful generalizations is an important part of statistical modeling. Introducing generalizations based only on the somewhat arbitrary criterion of infrequency captures some generalization but not as much, or as well, as adding sufficiently backed-off features. The backed off feature 2 in figure 8.4 furthermore, benefits by its more general nature from its incorporation of high-frequency instances such as feature 4 in the same set. The higher frequency of these features yields more reliable probabilities. Thus, feature 2's subsumption of feature 4 means that feature 2's frequency will be higher and more reliable than the merged feature 3.

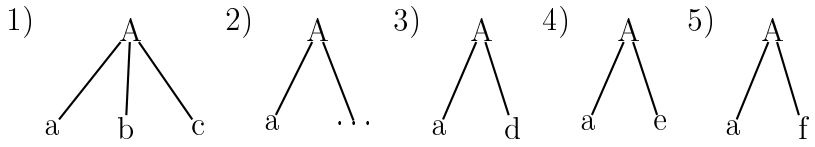

Figure 8.3: The feature set with another feature added.

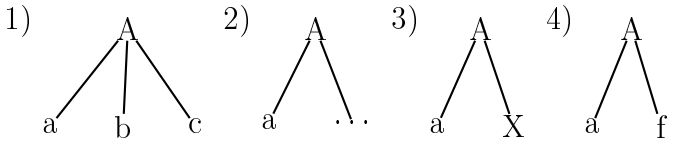

Figure 8.4: The new feature set with merged features. Note that unlike in figure 8.2, features 2 and 3 are not equivalent, since feature 4 is subsumed by feature 2, but not by feature 3.

All of these factors mean two things: firstly, the performance of the model should be enhanced by the inclusion of the less specific features, and secondly, the inclusion of these features will tend to make merged features redundant.

In principle, the justifications of using merging in conjunction with a cutoff are still well-founded. Merging creates more general features, whose frequencies are greater. If the merged features would have been lost in a cutoff but are retained due to the generalization, then it is true that non-noisy information is being saved by the model which would have been discarded. The problem is that if the overall feature set is well-designed, this saved information is not telling the model anything it does not already know.

It is worth noting that during early experiments on the Alvey tools data a brief attempt was made at merging rare features with more common features, thus increasing the total count of the new features. The results of this were predictably dismal, since the common features' reliability—a strength of the unmerged model—was compromised by lumping rarer and hence less reliable features in with them. The resultant merged features had the same weaknesses as merged features in the present experiments, as discussed in section 8.2, with the added disadvantage that they were replacing features which were contributing to successful modeling.

8.2.1 Selecting features to merge

The apparently negative results of chapter 7 might be regarded as inconclusive with respect to the usefulness of merging in general, since clearly not all possible combinations of elements were considered in the experiments. It is possible that there exists some approach to deciding upon elements to merge on which would yield better results. The experiments reported in this work centered entirely on

an approach based on the frequency of the elements. Indeed, with regard to a frequency-based feature cutoff, the possibility has been suggested that other criteria than frequency might be better employed as a basis for a cutoff, such as the variance of parameter values after training (Malouf, 2001). The possibility of merging according to different criteria than that used here, or searching the space of possible merges based upon some well-motivated heuristic remains open. Nevertheless, the problems illustrated in section 8.2 remain implicit in any strategy of merging on some subset of elements.

8.2.2 Modeling with rich feature sets

An early motivation behind attempting some compression-based approach such as merging on an overly specific feature set, rather than adding more general features to the model in advance, was the hope that through merging and a feature cutoff, an equivalent or near equivalent result could be gotten with a considerably smaller model. This intuition has proved to be somewhat misguided in several ways. For one, it disregards the fact that the more general features tend to be considerably more important to modeling than the specific features. In experiments with Alpino by other researchers, it has been reported to me that removing all features except rule feature still results in only slightly decreased performance, whereas removing only rule features results in performance near the baseline. From this it would appear that the best way to minimize the size of the model is to begin with the most general features and add specific features to them.

Another way in which the results of these experiments have shown the initial intuitions to be misguided is that the merging approach has proved to be most successful in cases of very small amounts of data and very sparse features. In fact, with the small size of such models, minimizing model size is not a major worry; adding a relatively small number of very general features will not increase the model size enough to matter, and will make an considerable difference in the performance.

8.3 Future directions

8.3.1 Modeling techniques

In order to do statistical modeling, a good supply of training data is crucial. For statistical parse selection, the training data is typically in the form of hand-parsed corpora, which are labor intensive and time consuming to build. Even in English, resources are limited and leave much to be desired. For other languages, good training data is even more difficult to come by. Ways around this have been sought, among them attempts at *unsupervised learning*, which trains on parsed,

unranked text and seeks to exploit patterns in the raw text to indicate underlying grammatical structure (Riezler et al., 2000). This work still uses a grammar, and thus may not qualify by some purist definitions as completely unsupervised, but no treebank is necessary. The advantage of such approaches would be less reliance upon difficult to obtain hand-parsed corpora; raw text can be easily obtained from the internet and other sources in large amounts and parsed as necessary by the grammar. So far, efforts at unsupervised training for parse selection have had some limited success, although they require vastly more data to achieve comparatively modest results. It appears at present that even an extremely small parsed corpus is better than none at all. An interesting direction for further research would be into combining supervised and unsupervised learning techniques to maximally exploit small training sets. Collins (2001) presents a framework in which various approaches to parameter setting may be compared, in order to investigate alternatives to modeling with maximum-likelihood variants (of which maxent is one). A number of algorithms from the field of machine learning are investigated in this paper, among them are a variant of the perceptron algorithm of Rosenblatt (1958), Support Vector Machines (SVMs) (Vapnik, 1998), and the AdaBoost algorithm of Freund and Schapire (1997). These approaches make different assumptions about what is known about the underlying distribution of the data than those made in maximum-likelihood modeling, and Collins argues that they may be more appropriate for the task of statistical parsing. In experiments applying SVM modeling techniques to parse selection, Dijkstra (2001) reports somewhat worse results than those reported by Osborne (2000a), who uses maxent modeling on a comparable data set, but work in this area is far from conclusive. Alternate approaches such as these would appear to warrant further investigation.

8.3.2 Smoothing operations

This thesis has been concerned with feature merging as an operation on models designed to reduce noise within the model. The primary other operation which it has been compared to here has been a frequency-based feature cutoff. Alternatives to these approaches are possible to imagine which might work better; feature cutoffs might be employed based on criteria other than frequency, for example standard deviation. Alternately, an operation might work on other aspects of the model than single features, such as, for example, the dependencies between individual features. Taking a cue from Pedersen, Bruce, and Wiebe (1997) and the work there on searching various complexity levels of graphical models, experiments might be carried out in which graph edges (dependencies) were deleted rather than nodes (features). Another approach to this, closer to the spirit of feature merging presented in this thesis, might be to establish an upper bound on the degree of dependency of two features, above which the features would be unioned, or "merged."

8.3.3 Parameter estimation

The IIS algorithm used in this thesis is one of several possible algorithms which may be employed to estimate the parameters of maxent models. Recent preliminary experiments conducted on the Alpino data by colleagues at the University of Groningen using the conjugate gradient (CG) algorithm (Shewchuck, 1994) suggest that this algorithm is worth further investigation. In particular, it has appeared to converge much more quickly on an optimal model with lower KL divergence than that converged on by IIS. Since the CG algorithm yields a model which appears to closer reflect the empirical distributions, overfitting has been observed to set in to a degree that is not found in similar models trained with IIS. This heightened sensitivity to empirical frequencies suggests that perhaps the merging techniques discussed in this thesis would provide some benefit to models trained in this way. It seems likely that any potential gains to be made in this way would be small, but possibly significant.

8.3.4 Search speed

Another difficulty encountered was the slowness with which it was possible to run experiments on models and come up with interesting results. Very recent preliminary experiments using the CG algorithm suggest that much greater training speeds may be achieved through use of alternate unconstrained optimization algorithms. It would be interesting to attain speeds at which it would be reasonable to compare large numbers of models rapidly, in order to make an exhaustive search for optimal cutoff levels and merge thresholds. The search for such values in the experiments reporte here was done by hand, and much finer-grained analysis would have been too time-consuming to consider. This, combined with the greater model sensitivity which might be attainable with alternate optimization algorithms suggests that operations along the lines of feature merging may be put to practical use yet, even in cases of models which appear relatively smooth when estimated using IIS.

8.3.5 Parsing in practice

It should go without saying that a specific task in an engineering field such as NLP should be defined appropriately with respect to the bigger picture of what, exactly, the goal of the system is. In many real-world cases, for such applications as information retrieval or question-answering systems, relatively shallow parsing strategies, which do not seek to assign a complete, detailed structure to sentences, are sufficient. Approaches to most NLP applications manage to bypass the need for a deep analysis of unrestricted text in a wide variety of ways; "controlled languages," restricted subsets of natural languages, are used in specific domains to aid machine translation, while keyword recognition, "chunking" and

shallow parsing are employed usefully in many applications. As mentioned in the introduction to this thesis, the work here in deep parsing straddles the line somewhat between purely applied NLP and linguistic investigation. In practice, deep parsing of unrestricted text, at current levels of performance, is not widely employed in NLP applications.

Part of the reason for this lies in the phrase "current levels of performance." It is easy to imagine applications which would benefit from high-performance, robust, deep parsing of unrestricted text. Machine translation applications, and natural language interfaces of all kinds would certainly benefit. One question, however, is whether the linguistic representations of language assumed today are capable of *ever* attaining the level of performance necessary to make this kind of parsing useful. As alluded to in the introduction and discussed in chapter 2, humans bring a great deal of semantic and real-world knowledge to bear in their own parsing process. It is not clear how, or whether, the amount and variety of information available to humans for natural language processing can every be usefully represented in the framework of a computational grammar. To put it another way, it is not at all clear the extent to which language may be employed by entities which do not know what they are saying. To this end, it will be interesting to follow research as it develops in knowledge representation and approaches to disambiguation which marry statistics with semantics. Clearly major challenge in this regard is the design and construction of appropriate corpora. Nevertheless, the idea of computational grammars incorporating information from semantic networks such as WordNet provides intriguing possibilities for statistical parsing.